A Balanced Memory-Based Collaborative Filtering Similarity Measure

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, Ángel Arroyo

Collaborative filtering recommender systems contribute to alleviating the problem of information overload that exists on the Internet as a result of the mass use of Web 2.0 applications. The use of an adequate similarity measure becomes a determining factor in the quality of the prediction and recommendation results of the recommender system, as well as in its performance. In this paper, we present a memory-based collaborative filtering similarity measure that provides extremely high-quality and balanced results; these results are complemented with a low processing time (high performance), similar to the one required to execute traditional similarity metrics. The experiments have been carried out on the MovieLens and Netflix databases, using a representative set of information retrieval quality measures.

1. INTRODUCTION

Recommender systems (RS) provide a relevant tool, which helps to mitigate part of the information overload generated via the use of Web 2.0 applications. RS provide personalized recommendations to users about items (books, music, films, gadgets, holiday destinations, etc.).[1–7]

The following types of filtering are used by RS:

- Content-based filtering[4,6]: The recommendations are based on the users' past choices (i.e., recommendation of a new programming book to a user who purchased various books on this subject the previous year).
- Demographic filtering[8]: The recommendations are based on the information provided by users considered similar according to demographic parameters, such as age, gender, nationality, and so on.

- Collaborative filtering (CF)[9–12]: The recommendations for each user (active user) are obtained in line with the preference of other users who have rated the products (items) in a similar way to the active user.
- Hybrid filtering[13]: The recommendations are made by combining the previous filtering; in particular, *content-based filtering/CF* and *demographic filtering/CF* are used.

Among the three types of basic filtering (content-based filtering, demographic filtering, CF), CF is the one that provides the best results. CF calculates the recommendations based on the information of the votes that all users have cast as regards their preferences on the items (i.e., in a film RS, the total preferences made by each user on each of the films they have voted for).

When the CF is solely based on the information stored in the array of votes, it is called memory-based CF.[9,11,14] A variety of CF exists which obtains information from additional sources to the array of votes, such as the social relations between users or the contents of posts in blogs; in these cases (memory-based + additional information) the additional information is used to improve the quality of the recommendations, but its use is only applicable to the subset of RS where that type of additional information exists.

All scientific progress in the area of memory-based CF has the virtue of being applicable in all types of CF-based RS (pure CF, *hybrid filtering*, memory-based + additional information); this is why the paper focuses on improving the results using memory-based CF.

The most commonly used algorithm to obtain predictions and recommendations founded on memory-based CF is the *K Nearest Neighbors* (KNN). In this case, to recommend items to a user (active)

- It searches for the most similar users (neighbors): This phase has a decisive impact on the quality of the predictions and recommendations provided. It is usually implemented using similarity metrics and measures between users (user–user CF). The paper focuses on this aspect of CF.
- It identifies the items most highly rated by their neighbors which have not been rated by this active user. The predictions are made by combining the neighbors' votes; for this purpose, it uses an *aggregation approach,*[9,12] such as *weighted sum* or *deviation from mean.*

Traditionally, the similarity metrics and measures used in RS come from those used in the statistics area or some of those used in various fields of information retrieval, such as Pearson correlation, cosine, adjusted cosine, and Spearman rank correlation.[9,11] Recent studies have shown that it is possible to improve the quality of the prediction and recommendation results[9,12,15] by using new memory-based CF similarity metrics and measures[16–20] specifically designed to make the most of the special feature inherent to RS and its more complex operating modes, such as high levels of sparsity[21] and cold-start situations.[22,23]

The RS memory-based CF similarity measures published which currently give the best quality prediction and recommendation results are Jaccard mean squared differences (JMSD)[16] and singularities (SING).[17] In the different papers published, the quality measures put to the test in order to analyze the metrics and similarity measures are primarily: MAE (Mean Absolute Error), root-mean-square error, coverage, precision, recall, and F1.

The paper is structured in the following way: Section 2 presents the motivation and the fundamental objective to be achieved; Section 3 explains the method followed to design the similarity measure; Section 4 shows the experiments carried out and the results obtained; finally, Section 5 sets out the most relevant conclusions.

2. OBJECTIVE

The JMSD metric[16] provides a high level of accuracy, precision, recall, and processing time performance, which together make it greatly superior to previous similarity measures. The SING metric[17] was developed after JMSD, whereas JMSD only analyzes the votes of each pair of users to be compared, SING analyzes the votes of each pair of users to be compared and, in addition, uses the information of the votes cast by the rest of the users.

SING achieves the same quality of accuracy, precision, and recall results as JMSD and, in addition, it significantly improves the coverage. This allows the enhancement of prediction and recommendation possibilities to users, which is particularly appreciated by users who request a large number of predictions and by those who are not satisfied with a limited number of recommendations.

The great drawback of SING lies in its processing time performance: SING needs to calculate totals of relevant and nonrelevant votes for each item of the database. This means that these totals must be obtained for each similarity calculation between users, or it must store the totals and update them on a regular basis.

The objective pursued is to design a similarity measure capable of combining the positive results of both metrics, and at the same time of reducing their drawbacks (bad coverage using JMSD and bad performance using SING).

3. METHOD

The first attempts to design the new metric consisted in creating hybrids of JMSD and SING. Unfortunately, none of the experiments provided the desired results when restricting the execution time to those of the traditional metrics. Finally, satisfactory results were achieved by focusing on improving the coverage of JMSD without using the design of SING.

JMSD is a metric with a very simple formulation. Its motivation and foundations can be found in Ref. 16. JMSD combines (i) numerical similarity information between the two users compared (mean squared differences [MSD]) and (ii) nonnumerical information (structural information) of similarity between the two users compared (Jaccard).

Let I be the set of items of the RS and $r_{u,i}$ the vote of user u on item i.

Let $I_u = \{i \in I | r_{u,i} \neq \bullet\}$ be the set of items voted by user u.

$$\text{Jaccard}_{x\to y} = \text{Jaccard}_{y\to x} = \frac{|I_x \cap I_y|}{|I_x \cup I_y|} \tag{1}$$

$$\mathrm{MSD}_{x\to y}=\mathrm{MSD}_{y\to x}=\frac{\sum_{i\in I_x\cap I_y}\left(r_{x,i}-r_{y,i}\right)^2}{\left|I_x\cap I_y\right|} \tag{2}$$

$$\begin{aligned}&\mathrm{JMSD}_{x\to y}\ =\mathrm{JMSD}_{y\to x}=\mathrm{Jaccard}_{x\to y}\left(1-\mathrm{MSD}_{x\to y}\right),\\ &r_{u,i}\in[0,1]\Rightarrow \mathrm{JMSD}_{x\to y}\in[0,1]\end{aligned} \tag{3}$$

The proposed metric adds a term to the JMSD equation which positively rates the coverage that user x can provide to user$y$ (4); basically, it counts the proportion of items that y has not voted and that x has, which are the items in which x can offer a prediction to active user y.

$$\mathrm{Coverage}_{x\to y}=\frac{\left|I_x-I_y\right|}{|I|},\quad y\equiv\text{active user},\quad \mathrm{Coverage}_{x\to y}\neq\mathrm{Coverage}_{y\to x} \tag{4}$$

The proposed similarity measure, which we will call CJMSD (coverage + Jaccard + MSD), combines the coverage factor which is contributed by term (4), with the similarity factor which is contributed by term (2), with the coincidence factor which is contributed by Jaccard (1):

$$\mathrm{CJMSD}_{x\to y}=\frac{\left|I_x-I_y\right|}{|I|}\frac{\left|I_x\cap I_y\right|}{\left|I_x\cup I_y\right|}\frac{\sum_{i\in I_x\cap I_y}\left[1-\left(r_{x,i}-r_{y,i}\right)^2\right]}{\left|I_x\cap I_y\right|} \tag{5}$$

Finally,

$$\begin{aligned}&\mathrm{CJMSD}_{x\to y}=\frac{\left|I_x-I_y\right|\sum_{i\in U_x\cap U_y}\left[1-\left(r_{x,i}-r_{y,i}\right)^2\right]}{|I|\left|I_x\cup I_y\right|},\\ &y\equiv\text{active user},\quad r_{u,i}\in[0,1]\Rightarrow \mathrm{CJMSD}_{x\to y}\in[-1,1]\end{aligned} \tag{6}$$

Note that, in general, $\mathrm{CJMSD}_{x\to y}\neq\mathrm{CJMSD}_{y\to x}$, and that this similarity measure makes it possible for a user x (neighbor candidate), who has cast many votes, to be neighbor of a user y (active user) who has cast few votes; this strengthens the prediction capacity. On the other hand, this similarity measure complicates the fact that y belongs to the neighborhood set of x, due to the fact that the prediction capacity of$y$ over x, is generally very low.

4. RESULTS

With the aim of testing the proposed similarity measure, a series of experiments have been carried out using the databases MovieLens 1M (4382 users, 3952 movies,

Table I. Main parameters used in the experiments and figures where its results are displayed.

	K (MAE, coverage)		Precision/recall			Common parameters		
	Range	step	*N*	θ	*K*	Test users (%)	Test items (%)	Figures
Movielens 1M	{50,. . . ,1000}	50	{2,. . . ,20}	5	200	20	20	1
Netflix	{50,. . . ,1000}	50	{2,. . . ,50}	5	200	5	20	2

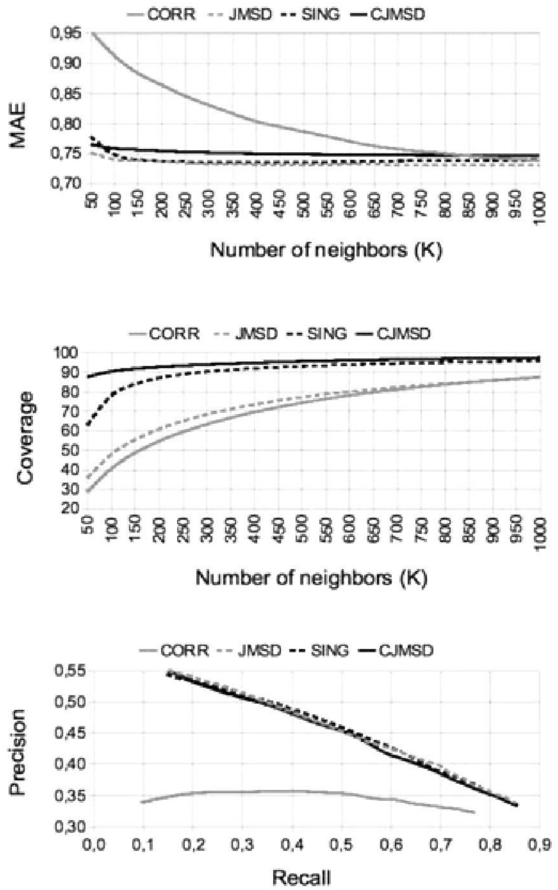

Figure 1. Results using MovieLens.

1,000,209 ratings) and Netflix (480,189 users, 17,770 items, 100,480,507 ratings). In these experiments, we obtain the quality measures: MAE, coverage, precision versus recall, processing time. Table I shows the main parameters used.

The similarity measure proposed (CJMSD) is compared with the two reference metrics (JMSD and SING) and with Pearson correlation (CORR) as a baseline.

Both the results obtained using Movielens 1M as the database (Figure 1), and the results obtained using Netflix as the database (Figure 2), show us that the

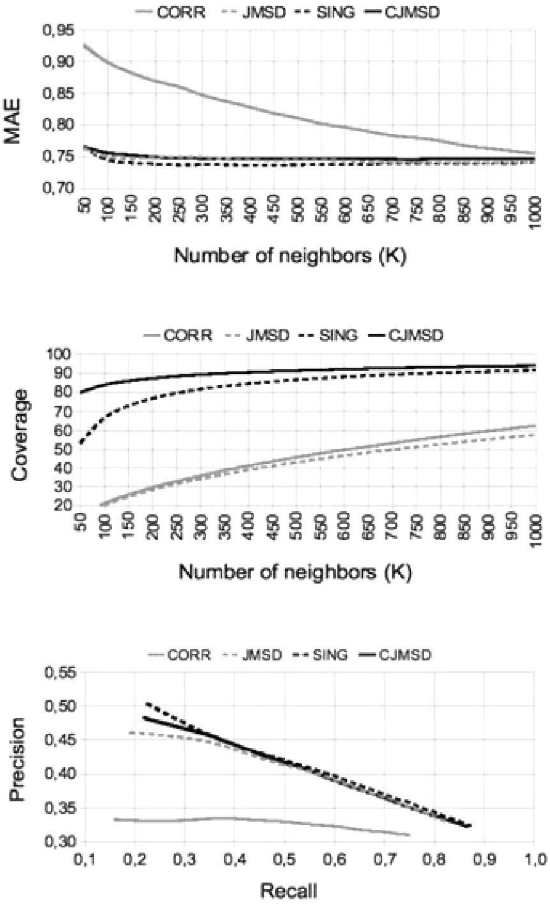

Figure 2. Results using Netflix.

objective pursued (Section 2) is met as regards the prediction and recommendation quality measures:

- The MAE of CJMSD is only slightly higher (worst) than the MAE of JMSD and SING.
- The quality of the precision and recall obtained with CJMSD is similar to that obtained using JMSD and SING.
- The coverage of CJMSD is even higher to that provided by SING.

Table II shows the processing times required for each of the similarity measures compared. The times indicated correspond to the average number of seconds needed to calculate the similarities of the test users with the training users 100 times (SING includes the calculations of the total number of relevant and nonrelevant votes for each item of the database).

By analyzing Table II, we can determine that the metric designed (CJMSD) complies with the target performance restrictions required (Section 2): the processing

Table II. Required processing time for each similarity measure.

	CORR	JMSD	SING	CJMSD
Processing time (s)	1.08	1.02	25.16	1.11

time of CJMSD is similar to the processing time of JMSD, and, therefore, much less than that required by SING.

5. CONCLUSIONS

The RS memory-based collaborative filtering metrics published which experimentally achieve the best prediction and recommendation results are JMSD and SING. However, each of these presents an aspect it would be advisable to improve: the coverage of JMSD and the processing time of SING. The proposed similarity measure (CJMSD) manages to broadly improve these two aspects, at the expense of very slightly worsening the most positive characteristics of JMSD: its MAE and its processing time.

The similarity measure presented in the paper provides very high and balanced quality results, which are complemented with a low processing time (high performance), similar to that required to execute the traditional similarity metrics.

As future work, we propose: (1) to test the behavior of the proposed similarity measure in new user cold-start situations. We believe that the obtained increase in coverage will result in an increase in the capacity to predict of the neighborhood selected for each cold-start user, (2) to use the similarity measure proposed to make recommendations to groups of users. In this situation, it is sometimes difficult to find recommendable items for the full group of users; the increase in the coverage obtained promises to improve this aspect.

Acknowledgment

Our acknowledgement to the Grouplens Research Group and to the FilmAffinity.com and Netflix companies. The authors thankfully acknowledges the computer resources, technical expertise, and assistance provided by the Supercomputation and Visualization Center of Madrid (CesViMa) and to the Spanish Supercomputing Network.

References

1. Serrano J, Viedma EH, Olivas JA, Cerezo A, Romero FP. A Google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0. Inf Sci 2011;181(8):1503–1516.
2. Li SH, Myaeng B, Kim M. A probabilistic music recommender considering user opinions and audio features. Inf Process Manag 2007;43(2):473–487.
3. Porcel C, Herrera-Viedma E. Dealing with incomplete information in a fuzzy linguistic_recommender system to disseminate information in university digital libraries. Knowl-Based Syst 2010;23(1):32–39.

4. Antonopoulus N, Salter J. Cinema screen recommender agent: combining collaborative and content-based filtering. IEEE Intell Syst 2006; 35–41.
5. Bobadilla J, Serradilla F, Hernando A. Collaborative filtering adapted to recommender systems of e-learning. Knowl-Based Syst 2009; 22:261–265.
6. Pudota N, Dattolo A, Baruzzo A, Ferrara F, Tasso C. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. Int J Intell Syst 2010;25(12):1158–1186.
7. Morales JM, Peis E, Ruiz AA, Herrera-Viedma A. Recommending biomedical resources: a fuzzy linguistic approach based on semantic Web. Int J Intell Syst 2010;25(12):1143–1157.
8. Vozalis MG, Margaritis KG. Using SVD and demographic data for the enhancement of generalized_collaborative filtering. Inf Sci 2007;177(15):3017–3037.
9. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 2005;17(6):734–749.
10. Shambour Q, Lu J. A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services. Int J Intell Syst 2011;26(9):814–843.
11. Herlocker JL, Konstan JA, Riedl JT, Terveen LG. Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 2004;22(1):5–53.
12. Bobadilla J, Hernando A, Ortega F, Bernal J. A framework for collaborative filtering recommender systems. Expert Syst Appl 2011;38(12):14609–14623.
13. Porcel C, Tejeda-Lorente A, Martínez MA, Herrera-Viedma E. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. Inf Sci 2011;184(1):1–19.
14. Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conf Uncertain Artif Intelligence, Morgan Kaufmann, Madison, WI, USA; 1998. pp 43–52.
15. Hernández F, Gaudioso E. Evaluation of recommender systems: a new approach. Expert Syst Appl 2008;35(3):790–804.
16. Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behavior of recommender systems. Knowl-Based Syst 2010;23:520–528.
17. Bobadilla J, Ortega F, Hernando A. A collaborative filtering similarity measure based on singularities. Inf Process Manag 2011;48(2):204–217.
18. Bobadilla J, Hernando A, Ortega F, Gutiérrez A. Collaborative filtering based on significances. Inf Sci 2012;185(1):1–17.
19. Bobadilla J, Ortega F, Hernando A, Bernal J. Generalization of recommender systems: collaborative filtering extended to groups of users and restricted to groups of items. Expert Syst Appl 2012;39(1):172–186
20. Bobadilla J, Ortega F, Hernando A, Alcalá J. Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowl-Based Syst 2011;24(8):1310–1316.
21. Anand D, Bharadwaj KK. Utilizing various sparsity measures for enhancing accuracy of_collaborative recommender systems based on local and global similarities. Expert Syst Appl 2011;38(5):5101–5109.
22. Ahn HJ. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci 2008;178(1):37–51.
23. Bobadilla J, Ortega F, Hernando A, Bernal J. A collaborative filtering approach to mitigate the new user cold start problem. Knowl-Based Syst 2011;26:225–238.